

JavaScript 进阶爱前端题库

注意：

①undefined 可简写为 u ; true 可简写为 t ; false 可简写为 f ;

② 如果你认为程序会报错，请写“报错”2 字即可，不需指明错误原因。

③ 在答题卡上作答。

1. 程序有 3 行控制台输出，试依次写出。

```
console.log(a);
var a = 8;
function fn(){
    console.log(a);
    var a = 15;
}
fn();
console.log(a);
```

2. 程序有 3 行控制台输出，试依次写出。

```
console.log(a);
var a = 8;
function fn(){
    console.log(a);
    a = 15;
}
fn();
console.log(a);
```

3. 程序有 1 行控制台输出，试写出。

```
var a = true;
function fn(){
    if(!a){
        var a = 10;
    }
    console.log(a);
}
fn();
```

```
var a = 8;
function fn(){
    if(!a){
        var a = 15;
    }
    console.log(a);
}
fn();
```

5. 程序有 3 次弹窗，试依次写出。

```
var n = 0;
function a(){
    var n = 10;
    function b(){
        n++;
        alert(n);
    }
    b();
    return b;
}
var c = a();
c();
alert(n);
```

6. 程序有 3 次弹窗，试依次写出。

```
var a=4;
function b(x,y,a){
    alert(a);
    arguments[2]=10;
    alert(a);
}
a = b(1,2,3);
alert(a);
```

7. 程序有 3 行控制台输出，试依次写出。

```
var foo = "a";
(function(foo){
    console.log(foo);
    var foo = foo || "b";
    console.log(foo);
})(foo);
console.log(foo);
```

8. 程序有 4 次弹窗，试依次写出。

```
var a = 9;
function fn(){
    a = 0;
    return function(b){
        return b + a++;
    }
}
var f = fn();
var m = f(5);
alert(m);
var n = fn()(5);
alert(n);
var x = f(5);
alert(x);
alert(a);
```

9. 程序有 1 行控制台输出，试写出。

```
var a = 3;
var b = a = 4;
console.log(a === b);
```

10. 程序有 1 行控制台输出，试写出。

```
var obj1 = {};
var obj2 = {};
console.log(obj1 == obj2);
```

11. 程序有 1 行控制台输出，试写出。

```
var arr1 = ["a","b","c","d"];
var arr2 = arr1;
arr1.push(arr2.pop());
console.log(arr1 == arr2);
```

12. 程序有 1 行控制台输出，试写出。

```
var arr1 = ["a","b","c","d"];
var arr2 = arr1;
arr1.unshift(arr2.pop());
console.log(arr1 == arr2);
```

13. 程序有 1 行控制台输出，试写出。

```
var obj = {};
var fun = function(){};
alert(obj.constructor == fun.constructor);
```

14. 程序有 1 行控制台输出，试写出。

```
var obj = {};
var fun = function(){};
alert(fun.__proto__.__proto__ == obj.__proto__);
```

15. 程序有 5 行控制台输出，试写出。

```
console.log(Object instanceof Object);
console.log(Function instanceof Function);
console.log(Number instanceof Number);
console.log(String instanceof String);
console.log(Function instanceof Object);
```

16. 下面语句结果是 true 还是 false ?

```
(new Array()).constructor.constructor.constructor == (new Array()).constructor
```

17. 程序有 1 行控制台输出，试写出。

```
function Fun(){
    return 3;
}
var m = new Fun();
console.log(m == 3);
```

18. 程序有 2 行控制台输出，试写出。

```
function A(){
    this.m = 1;
}
function B(){
    this.m = 2;
}
A.call(B);
B.call(A);
var a = new A();
var b = new B();
console.log(a.m == B.m);
console.log(b.m == A.m);
```

19. 程序有 1 行控制台输出，试写出。

```
var obj = {
    "a" : function(){
        return this.b;
    },
    "b" : function(){
        return this.c;
    },
    "c" : 1
}
var result = obj.a();
console.log(result);
```

20. 程序有 1 行控制台输出，试写出。

```
function Fun(){
    this.a = 5;
    function fun(){
        this.a = 10;
    }
    fun.a = 15;
    return fun;
}
var o = new(new Fun());
console.log(o.a);
```

21. 程序有 4 行控制台输出，试依次写出。

```
function Fun(para1,para2,para3){
    var obj = {};
    obj.param1 = para1;
    obj.param2 = para2;
    obj.param3 = para3;
    return obj;
}
Fun.prototype.param4 = 4;
var o = new Fun(1,2,3);
console.log(o.param1);
console.log(o.param4);
console.log(o instanceof Fun);
console.log(o.__proto__ == Fun.prototype);
```

22. 程序有 1 行控制台输出，试写出。

```
function getLength(){
    return this.length;
}
function foo(){
    this.length = 1;
    return (function(){
        var length = 2;
        return {
            length : function(a,b,c){
                return this.arr.length
            },
            arr : [1,2,3,4],
            info : function(){
                return
            }
        }
    })();
}
var l = foo().info();
console.log(l);
```

23. 程序有 4 次弹窗，试依次写出。

```
function fun(f,a,b,c){
    arguments[0](5,6);
}
function fun2(p,q,r,s,t){
    alert(this.length);
    alert(this.callee.length);
    alert(arguments.length);
    alert(arguments.callee.length);
}
fun(fun2,8,9,10,11,12);
```

24. 程序有 3 次弹窗，试依次写出。

```
var Foo = (function(){
    var instance = null;
    return function(){
        if(instance == null){
            return instance = new Bar();
        }
        return instance;
    }
})();
var Bar = function(){}
var o1 = new Foo();
var o2 = new Foo();
alert(o1 === o2);
alert(o1 instanceof Foo);
alert(o2 instanceof Bar);
```

25. 程序有 5 行输出，试依次写出。

```
function A(){}
function B(){
    return new A();
}
A.prototype = B();
B.prototype = new B();
var a = new A();
var b = new B();
console.log(a.__proto__ == b.__proto__);
console.log(a instanceof A);
console.log(a instanceof B);
console.log(b instanceof A);
console.log(b instanceof B);
```

26. 程序有 5 行输出，试依次写出。

```
var number = 2;
var obj = {
    number : 4,
    fn1 : (function(){
        this.number *= 2;
        number = number * 2;
        var number = 3;
        return function(){
            this.number *= 2;
            number *= 3;
            alert(number);
        }
    })(),
    fn2 : function(){
        this.number *= 2;
    }
};
var fn1 = obj.fn1;
alert(number);
fn1();
obj.fn1();
obj.fn2();
alert(window.number);
alert(obj.number);
```